fc	fractional capability variable
$x,\ g,\ a,\ b$	expression variable
k	integer variable
el	array-element variable
l	location variable
M	matrix variable

m	$::=$			matrix expressions
	$\vert$	M		matrix variables
	$\vert$	$m + m'$		matrix addition
	$\vert$	$m\, m'$		matrix multiplication
	$\vert$	(m)	S	
f	$::=$			fractional capability
	$\vert$	fc		variable
	$\vert$	1		whole capability
	$\vert$	$\frac{1}{2} \cdot f$		
t	$::=$			linear type
	$\vert$	**unit**		unit
	$\vert$	**bool**		boolean (true/false)
	$\vert$	**int**		63-bit integers
	$\vert$	**elt**		array element
	$\vert$	f **arr**		arrays
	$\vert$	f **mat**		matrices
	$\vert$	$!t$		multiple-use type
	$\vert$	$\forall\, fc.t$	bind fc in t	frac. cap. generalisation
	$\vert$	$t \otimes t'$		pair
	$\vert$	$t \multimap t'$		linear function
	$\vert$	(t)	S	parentheses
p	$::=$			primitive
	$\vert$	**not**		boolean negation
	$\vert$	$(+)$		integer addition
	$\vert$	$(-)$		integer subtraction
	$\vert$	$(*)$		integer multiplication
	$\vert$	$(/)$		integer division
	$\vert$	$(=)$		integer equality
	$\vert$	$(\langle)$		integer less-than
	$\vert$	$(+.)$		element addition
	$\vert$	$(-.)$		element subtraction
	$\vert$	$(*.)$		element multiplication
	$\vert$	$(/.)$		element division
	$\vert$	$(=.)$		element equality
	$\vert$	$(<.)$		element less-than
	$\vert$	**set**		array index assignment
	$\vert$	**get**		array indexing
	$\vert$	**share**		share array
	$\vert$	**unshare**		unshare array
	$\vert$	**free**		free arrary
	$\vert$	**array**		Owl: make array
	$\vert$	**copy**		Owl: copy array
	$\vert$	**sin**		Owl: map sine over array
	$\vert$	**hypot**		Owl: $x_i := \sqrt{x_i^2 + y_i^2}$

	|	**asum**		BLAS: $\sum_i \lvert x_i \rvert$
	|	**axpy**		BLAS: $x := \alpha x + y$
	|	**dot**		BLAS: $x \cdot y$
	|	**rotmg**		BLAS: see its docs
	|	**scal**		BLAS: $x := \alpha x$
	|	**amax**		BLAS: $\mathrm{argmax}\, i : x_i$
	|	**setM**		matrix index assignment
	|	**getM**		matrix indexing
	|	**shareM**		share matrix
	|	**unshareM**		unshare matrix
	|	**freeM**		free matrix
	|	**matrix**		Owl: make matrix
	|	**copyM**		Owl: copy matrix
	|	**copyM_to**		Owl: copy matrix onto another
	|	**sizeM**		dimension of matrix
	|	**trnsp**		transpose matrix
	|	**gemm**		BLAS: $C := \alpha A^{T?} B^{T?} + \beta C$
	|	**symm**		BLAS: $C := \alpha AB + \beta C$
	|	**posv**		BLAS: Cholesky decomp. and solve
	|	**potrs**		BLAS: solve with given Cholesky
v	::=			values
	|	p		primitives
	|	x		variable
	|	$()$		unit introduction
	|	**true**		true
	|	**false**		false
	|	k		integer
	|	l		heap location
	|	el		array element
	|	**Many** v		!-introduction
	|	**fun** $fc \to v$		frac. cap. abstraction
	|	$v[f]$		frac. cap. specialisation
	|	(v, v')		pair introduction
	|	**fun** $x : t \to e$	bind x in e	abstraction
	|	**fix** $(g, x : t, e : t')$	bind $g \cup x$ in e	fixpoint
	|	(v)	S	parentheses
e	::=			expression
	|	p		primitives
	|	x		variable
	|	**let** $x = e$ **in** e'	bind x in e'	let binding
	|	$()$		unit introduction
	|	**let** $() = e$ **in** e'		unit elimination
	|	**true**		true
	|	**false**		false

$\mid$	$\textbf{if}\ e\ \textbf{then}\ e_1\ \textbf{else}\ e_2$		if
$\mid$	k		integer
$\mid$	l		heap location
$\mid$	el		array element
$\mid$	$\textbf{Many}\ e$		!-introduction
$\mid$	$\textbf{let Many}\ x = e\ \textbf{in}\ e'$		!-elimination
$\mid$	$\textbf{fun}\ fc \to e$		frac. cap. abstraction
$\mid$	$e[f]$		frac. cap. specialisation
$\mid$	(e, e')		pair introduction
$\mid$	$\textbf{let}\ (a, b) = e\ \textbf{in}\ e'$	bind $a \cup b$ in e'	pair elimination
$\mid$	$\textbf{fun}\ x : t \to e$	bind x in e	abstraction
$\mid$	$e\ e'$		application
$\mid$	$\textbf{fix}\ (g, x : t, e : t')$	bind $g \cup x$ in e	fixpoint
$\mid$	(e)	S	parentheses

C	$::=$		evaluation contexts
$\mid$	$\textbf{let}\ x = [-]\ \textbf{in}\ e$	bind x in e	let binding
$\mid$	$\textbf{let}\ () = [-]\ \textbf{in}\ e$		unit elimination
$\mid$	$\textbf{if}\ [-]\ \textbf{then}\ e_1\ \textbf{else}\ e_2$		if
$\mid$	$\textbf{Many}\ [-]$		!-introduction
$\mid$	$\textbf{let Many}\ x = [-]\ \textbf{in}\ e$		!-elimination
$\mid$	$\textbf{fun}\ fc \to [-]$		frac. cap. abstraction
$\mid$	$[-][f]$		frac. cap. specialisation
$\mid$	$([-], e)$		pair introduction
$\mid$	$(v, [-])$		pair introduction
$\mid$	$\textbf{let}\ (a, b) = [-]\ \textbf{in}\ e$	bind $a \cup b$ in e	pair elimination
$\mid$	$[-]e$		application
$\mid$	$v[-]$		application

Θ	$::=$	fractional capability environment
$\mid$	$\cdot$	
$\mid$	Θ, fc	

Γ	$::=$	linear types environment
$\mid$	$\cdot$	
$\mid$	$\Gamma, x : t$	
$\mid$	Γ, Γ'	

Δ	$::=$	intuitionistic types environment
$\mid$	$\cdot$	
$\mid$	$\Delta, x : t$	

σ	$::=$	heap
$\mid$	$\{\}$	empty heap
$\mid$	$\sigma \uplus \{l \mapsto_f m_{k_1,k_2}\}$	location l points to matrix m

$\boxed{\Theta \vdash f\ \mathsf{Cap}}$ Valid fractional capabilities

$$\frac{fc \in \Theta}{\Theta \vdash fc\ \mathsf{Cap}}\ \textsc{WF\_Cap\_Var}$$

$$\frac{}{\Theta \vdash 1\ \mathsf{Cap}}\ \textsc{WF\_Cap\_Zero}$$

$$\frac{\Theta \vdash f\ \mathsf{Cap}}{\Theta \vdash \frac{1}{2} \cdot f\ \mathsf{Cap}}\ \textsc{WF\_Cap\_Succ}$$

$\boxed{\Theta \vdash t\ \mathsf{Type}}$ Valid types

$$\frac{}{\Theta \vdash \mathbf{unit}\ \mathsf{Type}}\ \textsc{WF\_Type\_Unit}$$

$$\frac{}{\Theta \vdash \mathbf{bool}\ \mathsf{Type}}\ \textsc{WF\_Type\_Bool}$$

$$\frac{}{\Theta \vdash \mathbf{int}\ \mathsf{Type}}\ \textsc{WF\_Type\_Int}$$

$$\frac{}{\Theta \vdash \mathbf{elt}\ \mathsf{Type}}\ \textsc{WF\_Type\_Elt}$$

$$\frac{\Theta \vdash f\ \mathsf{Cap}}{\Theta \vdash f\ \mathbf{arr}\ \mathsf{Type}}\ \textsc{WF\_Type\_Array}$$

$$\frac{\Theta \vdash t\ \mathsf{Type}}{\Theta \vdash !t\ \mathsf{Type}}\ \textsc{WF\_Type\_Bang}$$

$$\frac{\Theta, fc \vdash t\ \mathsf{Type}}{\Theta \vdash \forall fc.t\ \mathsf{Type}}\ \textsc{WF\_Type\_Gen}$$

$$\frac{\begin{array}{c}\Theta \vdash t\ \mathsf{Type}\\ \Theta \vdash t'\ \mathsf{Type}\end{array}}{\Theta \vdash t\ \otimes\ t'\ \mathsf{Type}}\ \textsc{WF\_Type\_Pair}$$

$$\frac{\begin{array}{c}\Theta \vdash t\ \mathsf{Type}\\ \Theta \vdash t'\ \mathsf{Type}\end{array}}{\Theta \vdash t\ \multimap\ t'\ \mathsf{Type}}\ \textsc{WF\_Type\_Lolly}$$

$\boxed{\Theta; \Delta; \Gamma \vdash e : t}$ Typing rules for expressions

$$\frac{}{\Theta; \Delta; \cdot, x : t \vdash x : t}\ \textsc{Ty\_Var\_Lin}$$

$$\frac{x : t\ \in\ \Delta}{\Theta; \Delta; \cdot \vdash x : t}\ \textsc{Ty\_Var}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash e : t\\ \Theta; \Delta; \Gamma', x : t \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let}\ x = e\ \mathbf{in}\ e' : t'}\ \textsc{Ty\_Let}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash () : \mathbf{unit}}\ \textsc{Ty\_Unit\_Intro}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \cdot \vdash e : \mathbf{unit}\\ \Theta; \Delta; \Gamma \vdash e' : t\end{array}}{\Theta; \Delta; \Gamma \vdash \mathbf{let}\ () = e\ \mathbf{in}\ e' : t}\ \textsc{Ty\_Unit\_Elim}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash \mathbf{true} : !\mathbf{bool}}\ \textsc{Ty\_Bool\_True}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash \mathbf{false} : !\mathbf{bool}}\ \textsc{Ty\_Bool\_False}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash e : \mathbf{bool}\\ \Theta; \Delta; \Gamma' \vdash e_1 : t'\\ \Theta; \Delta; \Gamma' \vdash e_2 : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{if}\ e\ \mathbf{then}\ e_1\ \mathbf{else}\ e_2 : t}\ \textsc{Ty\_Bool\_Elim}$$

$$\frac{}{\Theta;\Delta;\cdot\vdash k:\,!\mathbf{int}}\ \textsc{Ty\_Int\_Intro}$$

$$\frac{}{\Theta;\Delta;\cdot\vdash el:\,!\mathbf{elt}}\ \textsc{Ty\_Elt\_Intro}$$

$$\frac{\begin{array}{l}\Theta;\Delta;\cdot\vdash v:t\\ v\neq l\end{array}}{\Theta;\Delta;\cdot\vdash \mathbf{Many}\ v:\,!t}\ \textsc{Ty\_Bang\_Intro}$$

$$\frac{\begin{array}{l}\Theta;\Delta;\Gamma\vdash e:\,!t\\ \Theta;\Delta,x:t;\Gamma'\vdash e':t'\end{array}}{\Theta;\Delta;\Gamma,\Gamma'\vdash \mathbf{let\,Many}\ x=e\ \mathbf{in}\ e':t'}\ \textsc{Ty\_Bang\_Elim}$$

$$\frac{\begin{array}{l}\Theta;\Delta;\Gamma\vdash e:t\\ \Theta;\Delta;\Gamma'\vdash e':t'\end{array}}{\Theta;\Delta;\Gamma,\Gamma'\vdash (e,e'):t\otimes t'}\ \textsc{Ty\_Pair\_Intro}$$

$$\frac{\begin{array}{l}\Theta;\Delta;\Gamma\vdash e_{12}:t_1\otimes t_2\\ \Theta;\Delta;\Gamma',a:t_1,b:t_2\vdash e:t\end{array}}{\Theta;\Delta;\Gamma,\Gamma'\vdash \mathbf{let}\,(a,b)=e_{12}\ \mathbf{in}\ e:t}\ \textsc{Ty\_Pair\_Elim}$$

$$\frac{\begin{array}{l}\Theta\vdash t'\ \mathsf{Type}\\ \Theta;\Delta;\Gamma,x:t'\vdash e:t\end{array}}{\Theta;\Delta;\Gamma\vdash \mathbf{fun}\,x:t'\to e:t'\multimap t}\ \textsc{Ty\_Lambda}$$

$$\frac{\begin{array}{l}\Theta;\Delta;\Gamma\vdash e:t'\multimap t\\ \Theta;\Delta;\Gamma'\vdash e':t'\end{array}}{\Theta;\Delta;\Gamma,\Gamma'\vdash e\ e':t}\ \textsc{Ty\_App}$$

$$\frac{\Theta,fc;\Delta;\Gamma\vdash e:t}{\Theta;\Delta;\Gamma\vdash \mathbf{fun}\,fc\to e:\forall fc.t}\ \textsc{Ty\_Gen}$$

$$\frac{\begin{array}{l}\Theta\vdash f\ \mathsf{Cap}\\ \Theta;\Delta;\Gamma\vdash e:\forall fc.t\end{array}}{\Theta;\Delta;\Gamma\vdash e[f]:t[f/fc]}\ \textsc{Ty\_Spc}$$

$$\frac{\Theta;\Delta,g:t\multimap t';\cdot,x:t\vdash e:t'}{\Theta;\Delta;\cdot\vdash \mathbf{fix}\,(g,x:t,e:t'):\,!(t\multimap t')}\ \textsc{Ty\_Fix}$$

$\boxed{\langle\sigma,e\rangle\to\langle\sigma',e'\rangle}$ operational semantics

$$\frac{}{\langle\sigma,\mathbf{let}\,()=()\ \mathbf{in}\ e\rangle\to\langle\sigma,e\rangle}\ \textsc{Op\_Let\_Unit}$$

$$\frac{}{\langle\sigma,\mathbf{let}\ x=v\ \mathbf{in}\ e\rangle\to\langle\sigma,e[x/v]\rangle}\ \textsc{Op\_Let\_Var}$$

$$\frac{}{\langle\sigma,\mathbf{if\ true\ then}\ e_1\ \mathbf{else}\ e_2\rangle\to\langle\sigma,e_1\rangle}\ \textsc{Op\_If\_True}$$

$$\frac{}{\langle\sigma,\mathbf{if\ false\ then}\ e_1\ \mathbf{else}\ e_2\rangle\to\langle\sigma,e_2\rangle}\ \textsc{Op\_If\_False}$$

$$\frac{}{\langle\sigma,\mathbf{let\,Many}\ x=\mathbf{Many}\ v\ \mathbf{in}\ e\rangle\to\langle\sigma,e[x/v]\rangle}\ \textsc{Op\_Let\_Many}$$

$$\frac{e_1=e[g/\mathbf{let\,Many}\ g=\mathbf{fix}\,(g,x:t,e:t')\ \mathbf{in}\ \mathbf{fun}\,x:t\to e]}{\langle\sigma,\mathbf{let\,Many}\ g=\mathbf{fix}\,(g,x:t,e:t')\ \mathbf{in}\ e'\rangle\to\langle\sigma,e'[g/\mathbf{fun}\,x:t\to e_1]\rangle}\ \textsc{Op\_Let\_Fix}$$

$$\frac{}{\langle\sigma,(\mathbf{fun}\,fc\to v)[f]\rangle\to\langle\sigma,v[fc/f]\rangle}\ \textsc{Op\_Frac\_Cap}$$

$$\frac{}{\langle \sigma, (\textbf{fun}\, x : t \to e)\, v \rangle \to \langle \sigma, e[x/v] \rangle} \quad \textsc{Op\_App}$$

$$\frac{\langle \sigma, e \rangle \to \langle \sigma', e' \rangle}{\langle \sigma, C[e] \rangle \to \langle \sigma, C[e'] \rangle} \quad \textsc{Op\_Context}$$

$$\frac{0 \leq k_1, k_2}{\langle \sigma, \textbf{matrix}\, k_1\ k_2 \rangle \to \langle \sigma \uplus \{l \mapsto_1 M_{k_1,k_2}\}, l \rangle} \quad \textsc{Op\_Matrix}$$

$$\frac{}{\langle \sigma \uplus \{l \mapsto_1 M_{k_1,k_2}\}, \textbf{free}\, l \rangle \to \langle \sigma, () \rangle} \quad \textsc{Op\_Free}$$

$$\frac{}{\langle \sigma \uplus \{l \mapsto_f M_{k_1,k_2}\}, \textbf{share}\, l \rangle \to \langle \sigma \uplus \{l \mapsto_{\frac{1}{2}\cdot f} M_{k_1,k_2}\} \uplus \{l \mapsto_{\frac{1}{2}\cdot f} M_{k_1,k_2}\}, (l, l) \rangle} \quad \textsc{Op\_Share}$$

$$\frac{}{\langle \sigma \uplus \{l \mapsto_{\frac{1}{2}\cdot f} M_{k_1,k_2}\} \uplus \{l \mapsto_{\frac{1}{2}\cdot f} M_{k_1,k_2}\}, \textbf{unshare}\, l\ l \rangle \to \langle \sigma \uplus \{l \mapsto_f M_{k_1,k_2}\}, l \rangle} \quad \textsc{Op\_Unshare}$$

$$\frac{\sigma' = \sigma \uplus \{l_1 \mapsto_{fc_1} M1_{k_1,k_2}\} \uplus \{l_2 \mapsto_{fc_2} M2_{k_2,k_3}\}}{\langle \sigma' \uplus \{l_3 \mapsto_1 M1_{k_1,k_3}\}, \textbf{gemm}\, l_1\ l_2\ l_3 \rangle \to \langle \sigma' \uplus \{l_3 \mapsto_1 (M_1\ M_2 + M_3)_{k_1,k_3}\}, ((l_1, l_2), l_3) \rangle} \quad \textsc{Op\_Gemm}$$

$$\mathcal{V}[\![\mathbf{unit}]\!] = \{(\{\}, *)\}$$

$$\mathcal{V}[\![!\mathbf{bool}]\!] = \{(\{\}, true), (\{\}, false)\}$$

$$\mathcal{V}[\![!\mathbf{int}]\!] = \{(\{\}, n) \mid 2^{-63} \leq n \leq 2^{63} - 1\}$$

$$\mathcal{V}[\![!\mathbf{elt}]\!] = \{(\{\}, f) \mid f \text{aIEEEFloat64}\}$$

$$\mathcal{V}[\![\mathbf{Z}\,\mathbf{mat}]\!] = \{(\{l \mapsto_{m_W} \_\}, l)\}$$

$$\mathcal{V}[\![(\mathbf{S}f)\mathbf{mat}]\!] = \{(\sigma, l) \mid l \in \mathrm{dom}(\sigma) \wedge \sigma \star \sigma \in \pi_1[\mathcal{V}[\![f\mathbf{mat}]\!]]\}$$

$$\begin{aligned}\mathcal{V}[\![!(t' \multimap t'')]\!] = {} & \{(\{\}, \mathbf{Many}\, v) \mid (\{\}, v) \in \mathcal{V}[\![t' \multimap t'']\!]\} \\ & \cup \{(\{\}, \mathbf{fix}(g, x : t, e : t')) \mid \forall(\sigma', v') \in \mathcal{V}[\![t']\!].\ e[x/v][g/\mathbf{fun}\, x : t \to e_1] \in \mathcal{C}[\![t']\!]\} \\ & \text{where} e_1 = e[g/\mathbf{let}\,\mathbf{Many}\, g = \mathbf{fix}\,(g, x : t, e : t')\,\mathbf{in}\,\mathbf{fun}\, x : t \to e]\end{aligned}$$

$$\mathcal{V}[\![!t']\!] = \{(\{\}, \mathbf{Many}\, v) \mid t \notin \{\mathbf{bool}, \mathbf{int}, \mathbf{elt}, t' \multimap t''\} \wedge (\{\}, v) \in \mathcal{V}[\![t']\!]\}$$

$$\mathcal{V}[\![\forall fc.\ t]\!] = \{(\sigma, \forall fc.\ v) \mid \forall f.\ (\sigma, v[fc/f]) \in \mathcal{V}[\![t[fc/f]]\!]\}$$

$$\mathcal{V}[\![t' \otimes t'']\!] = \{(\sigma, \langle v', v'' \rangle) \mid \exists \sigma', \sigma''.\ (\sigma', v') \in \mathcal{V}[\![t]\!] \wedge (\sigma'', v'') \in \mathcal{V}[\![t'']\!] \wedge \sigma = \sigma' \star \sigma''\}$$

$$\mathcal{V}[\![t' \multimap t'']\!] = \{(\sigma, \mathbf{fun}\, x : t' \to e'') \mid \forall(\sigma', v') \in \mathcal{V}[\![t']\!].\ \sigma = \sigma' \star \sigma'' \text{defined} \Rightarrow (\sigma' \star \sigma'', e''[x/v']) \in \mathcal{C}[\![t'']\!]\}$$

$$\mathcal{C}[\![t]\!] = \{(\sigma_s, e) \mid \forall \sigma_r.\ \sigma_s \star \sigma_r \text{defined} \Rightarrow \exists \sigma_f, v, .(\sigma_s \star \sigma_r, e) \to^n (\sigma_f \star \sigma_r, v) \in \mathcal{V}[\![t]\!]\}$$

$$\mathcal{S}[\![\Delta; \cdot]\!]\theta = \{(\{\}, \delta) \mid \mathrm{dom}(\Delta) = (\delta) \wedge \forall x \in \mathrm{dom}(\Delta).\ (\{\}, \delta(x)) \in \mathcal{V}[\![\theta(t)]\!]\}$$

$$\mathcal{S}[\![\Delta; \Gamma, x : t]\!]\theta = \{(\sigma \uplus \sigma_x, \delta[x \mapsto v_x]) \mid (\sigma, \delta) \in \mathcal{S}[\![\Delta; \Gamma]\!] \wedge (\sigma_x, v_x) \in \mathcal{V}[\![\theta(t)]\!]\}$$

$$[\![\Theta; \Delta; \Gamma \vdash e : t]\!] = \forall \theta, \delta, \sigma.\ \mathrm{dom}(\Theta) = \mathrm{dom}(\theta) \wedge (\sigma, \delta) \in \mathcal{S}[\![\Delta; \Gamma]\!]\theta \Rightarrow \sigma(\delta(e)) \in \mathcal{C}[\![\theta(t)]\!]$$